SEL-82-306

ANNOTATED BIBLIOGRAPHY OF SOFTWARE ENGINEERING LABORATORY LITERATURE

NOVEMBER 1985

NASA

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

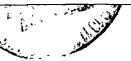# ANNOTATED BIBLIOGRAPHY OF SOFTWARE ENGINEERING LABORATORY LITERATURE

NOVEMBER 1985

NASA

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt. Maryland 20771

FOREWORD

The Software Engineering Laboratory (SEL) is an organization sponsored by the National Aeronautics and Space Administration, Goddard Space Flight Center (NASA/GSFC) and created for the purpose of investigating the effectiveness of software engineering technologies when applied to the development of applications software. The SEL was created in 1977 and has three primary organizational members:

    NASA/GSFC (Systems Development and Analysis Branch)

    The University of Maryland (Computer Sciences Department)

    Computer Sciences Corporation (Flight Systems Operation)

The goals of the SEL are (1) to understand the software development process in the GSFC environment; (2) to measure the effect of various methodologies, tools, and models on this process; and (3) to identify and then to apply successful development practices. The activities, findings, and recommendations of the SEL are recorded in the Software Engineering Laboratory Series, a continuing series of reports that includes this document. A version of this document was also issued as Computer Sciences Corporation document CSC/TM-85/6110.

The primary contributors to this document are

David Card	(Computer Sciences Corporation)
Q. L. Jordan	(Computer Sciences Corporation)
Frank McGarry	(Goddard Space Flight Center)

Single copies of this document can be obtained by writing to

Frank E. McGarry
Code 552
NASA/GSFC
Greenbelt, Maryland 20771

8978

ABSTRACT

This document is an annotated bibliography of technical papers, documents, and memorandums produced by or related to the Software Engineering Laboratory. More than 100 publications are summarized. These publications cover many areas of software engineering and range from research reports to software documentation.

This document has been updated and reorganized substantially since the original version (SEL-82-006, November 1982). All materials have been grouped into five general subject areas for easy reference:

- The Software Engineering Laboratory
- Software Tools
- Models and Measures
- Technology Evaluations
- Data Collection

An index further classifies these documents by specific topic.

8978   PAGE iv INTENTIONALLY BLANK

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

8978

TABLE OF CONTENTS (Cont'd)

8978

TABLE OF CONTENTS (Cont'd)

x

8978

TABLE OF CONTENTS (Cont'd)

8978

TABLE OF CONTENTS (Cont'd)

8978

TABLE OF CONTENTS (Cont'd)

8978

TABLE OF CONTENTS (Cont'd)

8978

TABLE OF CONTENTS (Cont'd)

8978

TABLE OF CONTENTS (Cont'd)

8978

8978

TABLE OF CONTENTS (Cont'd)

8978

TABLE OF CONTENTS (Cont'd)

8978

SECTION 1 - INTRODUCTION

This document is an annotated bibliography of technical papers, documents, articles, and memoranda produced by or related to the Software Engineering Laboratory (SEL). It is intended to provide a quick reference to the published results of SEL research and development activities.

More than 100 publications are summarized in this document. Each summary includes the size of the publication (number of pages), a description (abstract) of its contents, and its original citation. Previous versions and subsequent reprintings are also identified where appropriate.

The publications described here cover many aspects of software engineering and range from research reports to software documentation. They are divided into five general subject areas:

- The Software Engineering Laboratory
- Software Tools
- Models and Measures
- Technology Evaluations
- Data Collection

An index is included at the end of this document to assist in identifying materials related to specific topics. A list of the abstract numbers of related publications follows each subject heading in the index.

Copies of individual publications listed in this bibliography can be obtained from one or more of the sources shown in Table 1-1. The acronyms defined in the table appear after each abstract and indicate the document's availability. Any material not labeled with one of these acronyms can be obtained only from the author(s).

8978

Table 1-1.   Availability of SEL Literature

Acronym	Source	Address
FEM	F. E. McGarry	Code 582 Bldg. 23, Room E-245 Goddard Space Flight Center Greenbelt, Maryland 20771
NSTF	NASA Scientific and Technical Installation Facility[1] (and source above)	P. O. Box 8757 BWI Airport, Maryland 21240
NTIS	National Technological Information Service[2] (and sources above)	5285 Port Royal Road Springfield, Virginia 22161
JAO	Journals and other private publishers	See specific citation

[1] Open to Federal Government agencies only at no charge.

[2] There is a per-page charge for reprinting documents.

8978

SECTION 2 - THE SOFTWARE ENGINEERING LABORATORY

8978

2.1 COLLECTED SOFTWARE ENGINEERING PAPERS: VOLUME I, SEL-82-004, JULY 1982, 118 PAGES

This document is a collection of technical papers produced by participants in the SEL during the 5-year period ending December 31, 1981. The goal of the document is to make available, in one reference, some results of SEL research tnat originally appeared in a number of different forums. Although these papers cover a wide range of topics related to software engineering, they do not completely describe the activities and interests of the SEL.

For the convenience of this presentation, the 10 papers are organized into 4 major topics:

- SEL organization
- Resource models
- Software measures
- Software engineering applications

The SEL is still actively working to understand and improve the software development process at Goddard Space Flight Center (GSFC). Future efforts will be documented in subsequent volumes of the Collected Software Engineering Papers and in other SEL publications. FEM

2.2 COLLECTED SOFTWARE ENGINEERING PAPERS: VOLUME II, SEL-83-003, NOVEMBER 1983, 100 PAGES

This document is a collection of technical papers provided by SEL participants during the period January 1, 1982, to November 30, 1983. The objective of the document is to combine all research material originally appearing as several individual articles into one easy reference.

Although these papers cover a wide range of topics, they do not completely encompass the entire scope of the SEL activities or interests. For the convenience of this presentation, the nine papers are organized into four major topics:

- SEL organization
- Resource models
- Software measures
- Data collection

The SEL is still actively working to understand and improve the software development process at GSFC. Future efforts will be documented in subsequent volumes of <u>Collected Software Engineering Papers</u> and in other SEL publications. FEM

2.3 COLLECTED SOFTWARE ENGINEERING PAPERS: VOLUME III, SEL-85-003, NOVEMBER 1985, 132 PAGES

This document is a collection of technical papers provided by SEL participants during the period November 30, 1983, to November 1, 1985. The objective of the document is to combine all research material originally appearing as several individual articles into one easy reference.

Although these papers cover a wide range of topics, they do not completely encompass the entire scope of the SEL activities or interests. For the convenience of this presentation, the 12 papers are organized into 3 major topics:

- SEL organization
- Technology evaluations
- Software measurement

The SEL is still actively working to understand and improve the software development process at GSFC. Future efforts will be documented in subsequent volumes of Collected Software Engineering Papers and in other SEL publications. FEM

8978

2.4 GLOSSARY OF SOFTWARE ENGINEERING LABORATORY TERMS, SEL-82-105, T. A. BABST, M. G. ROHLEDER, AND F. E. MCGARRY, NOVEMBER 1983, 39 PAGES

This document is a glossary of terms used in the SEL. A list of acronyms is also included. The terms are defined within the context of the software development environment for flight dynamics at GSFC. The purposes of this document are to provide a concise reference for clarifying the language employed in SEL documents and data collection forms, establish standard definitions for use by SEL personnel, and explain basic software engineering concepts. A version of this document was also issued as Computer Sciences Corpora- tion document CSC/TM-83/6168. FEM

The previous version of this document was Glossary of Software Engineering Laboratory Terms, SEL-82-005, M. G. Rohleder, December 1982.

8978

2.5 NASA SOFTWARE RESEARCH AND TECHNOLOGY WORKSHOP (PROCEEDINGS), NATIONAL AERONAUTICS AND SPACE ADMINISTRATION, MARCH 1980, 139 PAGES

This document reproduces the presentations made by participants at the NASA Software Research and Technology Workshop held March 11 through 13, 1980, at NASA Headquarters in Washington, D. C. The overall intent of the workshop was to provide the basis for a coordinated agency-wide software research and technology program to meet agency requirements more efficiently and effectively. The topics of the workshop were

- Methodologies, tools, and techniques
- Programming languages
- Data base management software

SEL data collection methods were recommended to provide the quantitative information necessary for evaluating entries in the three areas above. Approximately 45 persons attended the workshop. FEM

8978

2.6 PROCEEDINGS FROM THE FIRST SUMMER SOFTWARE ENGINEERING WORKSHOP, SEL-76-001, AUGUST 1976, 194 PAGES

This document reproduces the presentations made by partici-
pants at the First Summer Software Engineering Workshop held
on August 5, 1976, at GSFC. The general topic of the con-
ference was software design. The presentations were grouped
into the following panels:

- Requirements analysis and design methodologies
- Program design languages
- Automated software tools

This first workshop surveyed available state-of-the-art
software development techniques. The specific applicability
of the techniques to the GSFC environment was also con-
sidered. Approximately 25 persons attended this meeting.
NTIS

2.7 PROCEEDINGS FROM THE SECOND SUMMER SOFTWARE ENGINEERING WORKSHOP, SEL-77-002, SEPTEMBER 1977, 146 PAGES

This document reproduces the presentations made by partici-
pants at the Second Summer Software Engineering Workshop
held on September 19, 1977, at GSFC. The presentations were
grouped into the following panels:

- Experimental design
- Models, measures, and metrics
- Data collection
- Software engineering experiences

This second workshop attempted to communicate with the
larger software engineering research community. Approaches
and experiences with the design of experiments and data col-
lection were reviewed. Approximately 55 persons attended
this meeting. NTIS

8978

2.8 PROCEEDINGS FROM THE THIRD SUMMER SOFTWARE ENGINEERING WORKSHOP, SEL-78-005, SEPTEMBER 1978, 132 PAGES

This document reproduces the presentations made by participants at the Third Summer Software Engineering Workshop held on September 18, 1978, at GSFC. The presentations were grouped into the following panels:

- The data collection process
- Validation of software development models
- Measuring software development methodologies
- Current activities and future directions

Many of the discussions at this third workshop dealt with "how" one collects software data and "how" one conducts successful software experiments. Approximately 70 persons attended this meeting. NTIS

8978

2.9 PROCEEDINGS FROM THE FOURTH SUMMER SOFTWARE ENGINEERING WORKSHOP, SEL-79-005, NOVEMBER 1979, 282 PAGES

This document reproduces the presentations made by participants at the Fourth Summer Software Engineering Workshop held on November 19, 1979, at GSFC. The presentations were grouped into the following panels:

- The Software Engineering Laboratory
- Data collection
- Experiments in methodology evaluation
- Software resource models
- Models and metrics of software development

This fourth workshop focused on actual experiences of data collection and the application of software methodologies, models, and tools. Approximately 100 persons attended this meeting. NTIS

8978

2.10 PROCEEDINGS FROM THE FIFTH ANNUAL SOFTWARE ENGINEERING WORKSHOP, SEL-80-006, NOVEMBER 1980, 242 PAGES

This document reproduces the presentations made by participants at the Fifth Annual Software Engineering Workshop held on November 24, 1980, at GSFC. The presentations were grouped into the following panels:

- The Software Engineering Laboratory
- Software cost/resource modeling
- Software reliability
- Measurement of the development process

This fifth workshop focused on actual experiences with the application of software methodologies and models. Approximately 140 persons attended this meeting. NTIS

8978

2.11 PROCEEDINGS FROM THE SIXTH ANNUAL SOFTWARE ENGINEERING WORKSHOP, SEL-81-013, DECEMBER 1981, 282 PAGES

This document reproduces the presentations made by partici-
pants at the Sixth Annual Software Engineering Workshop held
on December 2, 1981, at GSFC. The presentations were
grouped into the following panels:

- Evaluating software development characteristics
- Software metrics
- Software models
- Software methodologies

The proceedings also includes a summary of the presentations
and audience comments. This sixth workshop was an attempt
to gather the experiences of software developers in applying
modern programming practices and other software engineering
techniques. Approximately 200 persons attended this meet-
ing. NTIS

2.12 <u>PROCEEDINGS FROM THE SEVENTH ANNUAL SOFTWARE
ENGINEERING WORKSHOP</u>, SEL-82-007, DECEMBER 1982,
<u>400 PAGES</u>

This document reproduces the presentations made by partici-
pants at the Seventh Annual Software Engineering Workshop
held on December 1, 1982, at GSFC. The presentations were
grouped into the following panels:

- The Software Engineering Laboratory (SEL)
- Software tools
- Software errors
- Software cost estimation

The document also includes a summary of the presentations
and audience remarks. The major emphasis of the meeting was
on reporting and discussing actual experiences with software
methodologies, models, and tools. Approximately 250 people,
representing 9 universities, 22 agencies of the Federal
Government, and 43 private organizations, attended the meet-
ing. <u>NTIS</u>

8978

2.13 PROCEEDINGS FROM THE EIGHTH ANNUAL SOFTWARE ENGINEERING WORKSHOP, SEL-83-007, NOVEMBER 1983, 316 PAGES

This document reproduces the presentations made by participants at the Eighth Annual Software Engineering Workshop held on November 3, 1983 at GSFC. The presentations were grouped into the following panels:

- The Software Engineering Laboratory (SEL)
- Software testing
- Human factors in software engineering
- Software quality assessment

The document also includes a summary of the presentations and audience remarks. Approximately 250 people representing 5 universities, 23 agencies of the Federal Government, and 44 private companies attended the meeting. NTIS

2.14 PROCEEDINGS FROM THE NINTH ANNUAL SOFTWARE ENGINEERING WORKSHOP, SEL-84-004, NOVEMBER 1984, 349 PAGES

This document reproduces the presentations made by partici-pants at the Eighth Annual Software Engineering Workshop held on November 28, 1984, at GSFC. The presentations were grouped into the following panels:

- The Software Engineering Laboratory (SEL)
- Software error studies
- Experiments with software development
- Software tools

The document also includes a summary of the presentations and audience remarks. Approximately 300 people, repre-senting 7 universities, 26 agencies of the Federal Govern-ment, and 56 private companies, attended the meeting. NTIS

2.15 THE SOFTWARE ENGINEERING LABORATORY, SEL-77-001, V. R. BASILI, M. V. ZELKOWITZ, F. E. McGARRY, ET AL., MAY 1977, 112 PAGES

This document is the initial statement of SEL objectives and procedures. The basic objective of the SEL is to improve the quality of the software development process and its products in one specific environment. To do this, it is necessary to first establish a baseline for comparison by understanding the current process.

This document discusses the steps to be taken to establish this baseline, including software development factors to be studied, the types of experiments to be performed, and the methods of data collection to be used. The current (as of publication) status of research and development activities is also reviewed. The appendixes contain the following materials:

- Data collection forms
- Data base file formats
- Summaries of software projects studied

This document was also issued as University of Maryland Technical Report TR-535. NSTF

2.16 THE SOFTWARE ENGINEERING LABORATORY, SEL-81-104,
D. N. CARD, F. E. McGARRY, G. PAGE, ET AL.,
FEBRUARY 1982, 121 PAGES

This document describes the history, organization, operation, and research results of the SEL. The SEL is a joint effort of GSFC, Computer Sciences Corporation, and the University of Maryland. The objective of the SEL is to study and improve the software development process in the GSFC environment. The SEL has conducted extensive research in the following areas of software engineering:

- Methodology evaluation
- Tool evaluation
- Resource models
- Reliability models
- Software measures

The document outlines SEL efforts in these areas and presents some preliminary conclusions based on this work. The appendixes include descriptions of the software projects studied and summary statistics derived from this data. This document was also issued as CSC/TM-82/6033. NTIS

The previous version of this document was The Software Engineering Laboratory, SEL-81-004, D. N. Card, F. E. McGarry, G. Page, et al., September 1981.

8978

2.17 "THE SOFTWARE ENGINEERING LABORATORY: OBJECTIVES,"
V. R. BASILI AND M. V. ZELKOWITZ, PROCEEDINGS OF
THE FIFTEENTH ANNUAL CONFERENCE ON COMPUTER PERSONNEL
RESEARCH, AUGUST 1977, 14 PAGES

This technical paper provides an overview of the SEL and its
objectives. The original motivations for establishing the
SEL were the high cost of software development and the sub-
sequent need to optimize the development process. This
paper discusses the following aspects of the SEL with re-
spect to these motivations:

- Specific objectives of the SEL
- Software development factors to be investigated
- Data collection techniques
- Early SEL research activities

The importance of defining consistent software development
measures is a recurrent theme throughout the discussion. JAO

8978

2.18 "A SOFTWARE TECHNOLOGY EVALUATION PROGRAM," D. N. CARD, ANNAIS DO XVIII CONGRESSO NACIONAL DE INFORMATICA, OCTOBER 1985, 6 PAGES

This paper describes an ongoing technology evaluation program conducted by the SEL that is intended to resolve certain issues in the application of tools, practices, and techniques by software developers. A wealth of potentially beneficial software engineering tools, practices, and techniques has emerged in the past several years. Simultaneously, realization has grown that all software engineering technologies are not equally effective for all software development problems and environments. The steps to technology improvement include measurement, evaluation, and transference. The SEL collects measures on the production of FORTRAN software for spacecraft navigation systems. Recent SEL investigations demonstrated that the use of structured programming and quality assurance improves software reliability. Also, intensive computer use appears to be associated with low productivity. However, the major factor in both productivity and reliability continues to be personnel capability. Such technology evaluation programs provide an empirical basis for defining software development standards and selecting tools. JAO

8978

SECTION 3 - SOFTWARE TOOLS

3.1   "ANALYSIS SOFTWARE REQUIREMENTS FOR THE DATA RETRIEVAL
       SYSTEM," D. N. CARD AND V. E. CHURCH, COMPUTER SCIENCES
       CORPORATION, TECHNICAL MEMORANDUM, MARCH 1983, 24 PAGES

This technical memorandum describes the capabilities needed
by the Data Base Retrieval System (DARES) to support the
analysis of data from the SEL data base. A total of 15
analysis requirements are defined in the report. All the
requirements identified were developed in discussion with
GSFC, Computer Sciences Corporation, and University of
Maryland personnel experienced in the use of the data base.
The appendix to the report contains an operating scenario of
the Data Retrieval System. <u>FEM</u>

3.2 COMMON SOFTWARE MODULE REPOSITORY (CSMR) SYSTEM DE-SCRIPTION AND USER'S GUIDE, SEL-79-003, C. E. GOOREVICH, A. L. GREEN, AND S. R. WALIGORA, AUGUST 1979, 156 PAGES

This document is the system description and user's guide for the Common Software Module Repository (CSMR). The CSMR program is a software library utility that provides interactive access to a data base of software modules.

This document describes library control procedures, program capabilities, and operating procedures. The system design and individual module descriptions are also included. Various appendixes contain the data base file formats and the system implementation procedures. This document was also issued as Computer Sciences Corporation document CSC/SD-79/6103. FEM

3-3

3.3 "CONFIGURATION ANALYSIS TOOL DESIGN," F. K. BANKS, COMPUTER SCIENCES CORPORATION, TECHNICAL MEMORANDUM, MARCH 1980, 99 PAGES

This technical memorandum presents the design of a Configuration Analysis Tool (CAT). CAT is an interactive program that maintains a file of components under development and produces reports of software development activity. This memorandum includes the baseline diagram, system description, and detailed module descriptions and defines menus, displays, and file formats. FEM

3.4 CONFIGURATION ANALYSIS TOOL (CAT) SYSTEM DESCRIPTION AND USER'S GUIDE (REVISION 1), SEL-80-104, W. J. DECKER, DECEMBER 1982, 110 PAGES

This document is the system description and user's guide for a configuration management utility. The Configuration Analysis Tool (CAT) performs interactive recording and reporting of the status of software development and maintenance activities. CAT enhances the ability of a manager to monitor and control a large software development project.

This version of the document provides an updated description of the system organization, operating procedures, source listings, file formats, error messages, and implementation instructions. This document was also issued as Computer Sciences Corporation document CSC/SD-82/6125. FEM

The previous version of this document was Configuration Analysis Tool (CAT) System Description and User's Guide, SEL-80-004, F. K. Banks, W. J. Decker, J. G. Garrahan, et al., October 1980.

3.5 <u>COST AND RELIABILITY ESTIMATION MODELS (CAREM) USER'S</u>
     <u>GUIDE</u>, SEL-81-008, J. F. COOK AND E. EDWARDS,
     FEBRUARY 1981, 28 PAGES

This document describes the operation of an interactive
software cost and reliability modeling utility. The Cost
and Reliability Estimation Models (CAREM) program allows the
user to fit any of several common models to a selected sub-
set of SEL data. The following models are available in the
program:

- Doty
- GRC
- Tecolote
- Walston/Felix

This document includes a brief description of each model,
operating instructions, and sample sessions. This document
was also issued as a Goddard Space Flight Center technical
memorandum. <u>FEM</u>

3.6 FORTRAN STATIC SOURCE CODE ANALYZER DESIGN AND MODULE DESCRIPTIONS, SEL-78-001, E. M. O'NEILL, S. R. WALIGORA, AND C. E. GOOREVICH, FEBRUARY 1978, 98 PAGES

This document contains the system design and individual module descriptions of the FORTRAN Static Source Code Analyzer Program (SAP). This program is a source code scanning and statistical reporting program whose purpose is to collect data from actual software for studies of software measures by the SEL. The measures extracted by SAP include counts of the number of

- Source lines
- Comment lines
- Executable lines
- Decisions
- External references

These statistics are reported on a module-by-module basis and as final totals. This document was also issued as Computer Sciences Corporation document CSC/TM-78/6012. FEM

3.7 FORTRAN STATIC SOURCE CODE ANALYZER PROGRAM (SAP) SYSTEM DESCRIPTION, SEL-82-002, W. J. DECKER AND W. A. TAYLOR, AUGUST 1982, 227 PAGES

This document is the system description of the FORTRAN Static Source Code Analyzer Program (SAP). SAP is a software tool tnat scans FORTRAN source code and tabulates occurrences of specific program features, such as executable lines of code, decisions, and input/output statements.

This document defines the processing methods and the components of SAP. The appendixes fully describe the modules, COMMON blocks, and files. System generation procedures are also explained. Some important considerations for implementing SAP on computers other than the PDP-11/70 or the VAX-11/780 are discussed. This document was also issued as Computer Sciences Corporation document CSC/SD-82/6045. FEM

3.8 FORTRAN STATIC SOURCE CODE ANALYZER PROGRAM (SAP) USER'S GUIDE (REVISION 2), SEL-78-202, W. J. DECKER AND W. A. TAYLOR, APRIL 1985, 133 PAGES

This document presents the FORTRAN Static Source Code Analyzer Program (SAP) User's Guide (Revision 2). SAP is a software tool designed to assist SEL personnel in conducting studies of FORTRAN programs. SAP scans FORTRAN source code and produces reports that present statistics and measures of statements and structures that make up a module.

Revision 2 is the result of program modifications to update several reports and provide additional software quality analysis and recognition of all statements described in the FORTRAN 77 dialects of the host VAX-11/780 and IBM 4341 computers. It provides instructions for operating SAP and contains information useful in interpreting SAP output. The appendixes contain descriptions of the derivation of Halstead's and McCabe's measures. This document was also issued as Computer Sciences Corporation document CSC/SD-82/6044. NSTF

The previous versions of this document were FORTRAN Static Source Code Analyzer Program (SAP) User's Guide, SEL-78-002, E. M. O'Neill, S. R. Waligora, C. E. Goorevich, et al., February 1978, and FORTRAN Static Source Code Analyzer Program (SAP) User's Guide (Revision 1), SEL-78-102, W. J. Decker and W. A. Taylor, September 1982.

3.9 FUNCTIONAL REQUIREMENTS/SPECIFICATIONS FOR CODE 580 CONFIGURATION ANALYSIS TOOL (CAT), SEL-80-001, F. K. BANKS, A. L. GREEN, AND C. E. GOOREVICH, FEBRUARY 1980, 49 PAGES

This document presents the functional requirements and software specifications for a configuration management utility. The Configuration Analysis Tool (CAT) is an interactive program for recording and reporting the status of software development and maintenance activities. CAT was developed for use by flight dynamics software development projects.

This document describes the hardware and operating system configuration, software system functions, and operational environment. The appendix contains data set formats. This document was also issued as Computer Sciences Corporation document CSC/TM-80/6051. FEM

8978

3.10 SIMPL-D DATA BASE REFERENCE MANUAL, SEL-79-001, M. V. ZELKOWITZ, JULY 1979, 53 PAGES

This document describes a data base system for software engineering data implemented on the Univac 1100/42 computer at the University of Maryland. The data base contains flight dynamics development data collected by the SEL. The software is implemented in the SIMPL-D language. Data base files are sequentially organized. The document contains the following sections:

- Overview of the application area
- Description of the data base organization
- User's guide to data base functions
- Implementation considerations for the 1100/42
- Programmer's guide to data base maintenance

This document was also issued as University of Maryland Technical Report TR-788. FEM

3.11 SOFTWARE ENGINEERING LABORATORY (SEL) COMPENDIUM OF TOOLS (REVISION 1), SEL-81-107, W. J. DECKER, W. A. TAYLOR, AND E. J. SMITH, FEBRUARY 1982, 76 PAGES

This document contains a series of brief descriptions of software tools available on the SEL computers (a PDP-11/70 and a VAX-11/780). The tools described in this document support the following applications:

- Cost and resource modeling
- Configuration management
- Software library management
- Data base maintenance
- Document library indexing
- Financial reporting
- Requirements analysis
- Source code analysis
- Structured FORTRAN

.These brief descriptions help potential users to judge the suitability of these programs to their needs. This document was also issued as Computer Sciences Corporation document CSC/TM-82/6038. NTIS

The previous version of this document was Software Engineering Laboratory (SEL) Compendium of Tools, SEL-81-007, W. J. Decker, E. J. Smith, W. A. Taylor, et al., February 1981.

3.12 SOFTWARE ENGINEERING LABORATORY (SEL) DATA BASE MAINTE-NANCE SYSTEM (DBAM) USER'S GUIDE AND SYSTEM DESCRIPTION SEL-81-203, P. LO AND D. CARD, JUNE 1984, 253 PAGES

This document explains the organization and operation of the SEL Data Base Maintenance System (DBAM). The system provides the following functions:

- Create new data base files
- Archive data base on tape
- Restore data base from tape
- Update existing data base files

The document also describes the use of these functions, possible error messages, and operating restrictions. The appendixes contain baseline diagrams, component descriptions, task-build instructions, and file definitions. This version of the document describes the change report file and form as modified in April 1983. Updates to this document were added in June 1984 to include descriptions of new software added to the DBAM system to archive and restore data base files. This document was also issued as Computer Sciences Corporation document CSC/SD-83/6016UD2. FEM

The previous versions of this document were the Software Engineering Laboratory (SEL) Data Base Maintenance System (DBAM) User's Guide and System Description, SEL-81-003, September 1981, and SEL-81-103, July 1983.

8978

3.13 SOFTWARE ENGINEERING LABORATORY (SEL) DATA BASE ORGAN-
     IZATION AND USER'S GUIDE (REVISION 1), SEL-81-102,
     P. LO AND D. WYCKOFF, JULY 1983, 182 PAGES

This document describes the structure of the SEL data base.
The data base contains three types of files: header, form
data, and auxiliary files. The format and contents of each
file are described in detail. File access and programming
considerations are also discussed. The operation of some
data base reporting software is reviewed. The appendixes
include facsimiles of SEL data collection forms and a glos-
sary of software engineering terms. This version of the
document describes the change report file and form as mod-
ified in April 1983, and it contains a description of Data-
trieve record definitions of all file types as added in
March 1984. This document was also issued as Computer
Sciences Corporation document CSC/SD-83/6012UD2. NSTF

The previous version of this document was Software Engineer-
ing Laboratory (SEL) Data Base Organization and User's
Guide, SEL-81-002, D. C. Wyckoff, G. Page, and F. E. McGarry,
September 1981.

8978

3.14 <u>SOFTWARE ENGINEERING LABORATORY (SEL) DATA BASE REPORT-
ING SOFTWARE USER'S GUIDE AND SYSTEM DESCRIPTION</u>,
SEL-82-003, P. LO, AUGUST 1983, VOL. 1:   202 PAGES;
VOL. 2:   239 PAGES

This two-volume document presents the SEL data base report-
ing software user's guide and system description.   The SEL
data base reporting software programs provide formatted
listings and summary reports of the SEL data base contents.
This document is intended to serve as a reference or tool
for the SEL data base administrator, librarians, and pro-
grammers and for managers and researchers involved in SEL
data base activities.   It describes the operating procedures
and system information for 18 different reporting software
programs.

Volume 1 contains an introduction summarizing the reporting
software programs and detailed operating procedures for each
program.   Sample output reports from each program are pro-
vided.   Volume 2 contains descriptions of the structure and
functions of each reporting software program.   Baseline dia-
grams, module descriptions, and listings of program genera-
tion files are also included.   <u>FEM</u>

3.15   SOFTWARE ENGINEERING LABORATORY (SEL) DOCUMENT LIBRARY
       (DOCLIB) SYSTEM DESCRIPTION AND USER'S GUIDE,
       SEL-81-006, W. A. TAYLOR AND W. J. DECKER,
       DECEMBER 1981, 196 PAGES

This document presents tne system description and user's
guide for the Document Library (DOCLIB) support software.
The DOCLIB system consists of a set of programs that main-
tain and access a computer data base containing descriptions
of documents available from a library maintained by the SEL.
The maintenance program, LIBMGR, is used to add, delete, or
modify document descriptions. The access program, DOCLIB,
is used to select and display specific descriptions.

Documents can be selected by author, title fragment, refer-
ence number, organization, or subject. This document de-
scribes the system organization, operating procedures, file
structures, and system generation. This document was also
issued as Computer Sciences Corporation document
CSC/SD-81/6116.   FEM

3.16 STRUCTURED FORTRAN PREPROCESSOR (SFORT), SEL-77-003, B. CHU AND D. S. WILSON, SEPTEMBER 1977, 51 PAGES

This document describes the Structured FORTRAN Preprocessor (SFORT) used in the development of flight dynamics software. The use of SFORT extends the capabilities of the standard FORTRAN language and enables a user to write structured, top-down, label-free, FORTRAN-like code. SFORT converts this extended FORTRAN into standard FORTRAN, which can then be used as input to a standard FORTRAN compiler.

This document discusses the six additional statements that may be employed in conjunction with the standard FORTRAN statements. The additional statements are

- DOWHILE (FORTRAN logical expression)
- ENDDO
- .IF (FORTRAN logical expression)
- ELSE
- ENDIF
- INCLUDE member name

Examples of the use of these statements in coding, as well as the IBM S/360 job control language (JCL) necessary to execute the preprocessor, are included. This document was also issued as Computer Sciences Corporation document CSC/TM-77/6256. FEM

3.17 STRUCTURED FORTRAN PREPROCESSOR (SFORT) PDP-11/70 USER'S GUIDE, SEL-78-004, D. S. WILSON AND B. CHU, SEPTEMBER 1978, 19 PAGES

This document provides users of the Structured FORTRAN Pre-processor (SFORT) 11/70 with the information needed to write programs using SFORT constructs. SFORT extends the capabilities of the FORTRAN IV language and enables a user to write structured, top-down, label-free, FORTRAN-like code. SFORT converts this extended FORTRAN into standard FORTRAN, which can then be used as input to a compiler. The additional statements processed by SFORT include

- DOWHILE (FORTRAN logical expression)
- ENDDO
- .IF (FORTRAN logical expression)
- ELSE
- ENDIF
- INCLUDE member name

This document was also issued as Computer Sciences Corporation document CSC/SD-78/6128. NSTF

3-18

3.18   SOFTWARE ENGINEERING LABORATORY (SEL) DATA BASE
       RETRIEVAL SYSTEM (DARES) USER'S GUIDE, SEL-83-104,
       T. A. BABST, W. J. DECKER, P. LO, ET AL.,
       SEPTEMBER 1984, 281 PAGES

This document is the user's guide for the SEL Data Base Re-
trieval System (DARES).   DARES gives data base users two
types of access to the development project data: (1) re-
ports summarizing the data in the data base and (2) options
to extract and move selected data into a work area from
which user-defined or predefined reports and plots may be
produced.   This document contains an overview of DARES, a
description of the operation of DARES, and instructions for
incorporating user software into DARES.   Two appendixes
present menu and help displays and system messages.   A ver-
sion of this document was also issued as Computer Sciences
Corporation document CSC/SD-84/6035.   FEM

The previous version of this document was Software Engineer-
ing Laboratory (SEL) Data Base Retrieval System (DARES)
User's Guide, SEL-83-004, T. A. Babst and W. J. Decker,
November 1983.

8978

3.19 SOFTWARE ENGINEERING LABORATORY (SEL) DATA BASE
RETRIEVAL SYSTEM (DARES) SYSTEM DESCRIPTION,
SEL-83-105, P. LO, W. K. MILLER, AND W. J. DECKER,
AUGUST 1984, 373 PAGES

This document presents the SEL Data Base Retrieval System
(DARES) system description. DARES provides data base users
with two types of access to development project data:
(1) reports summarizing the data in the data base and
(2) options to extract and move selected data into a work
area from which user-defined or predefined reports and plots
may be produced. This document contains an overview of
DARES; a description of the processing performed; and de-
scriptions of subroutines, COMMON blocks, file layouts, and
system generation procedures. This document was also issued
as Computer Sciences Corporation document CSC/SD-84/6034.

FEM

The previous version of this document was Software Engineer-
ing Laboratory (SEL) Data Base Retrieval System (DARES)
System Description, SEL-83-005, P. Lo and W. J. Decker,
November 1983.

8978

3.20 "ARROWSMITH-P--A PROTOTYPE EXPERT SYSTEM FOR SOFTWARE ENGINEERING MANAGEMENT," V. R. BASILI AND C. L. RAMSEY, PROCEEDINGS OF THE IEEE/MITRE EXPERT SYSTEMS IN GOVERNMENT SYMPOSIUM, OCTOBER 1985, 13 PAGES

This paper evaluates two prototype expert systems, collectively named ARROWSMITH-P. ARROWSMITH-P is intended to aid the manager of a software development project in an automated manner. The systems work as follows. First, it is determined whether a software project is following normal development patterns by comparing measures such as programmer hours per line of source code against historical, environment-specific baselines of such measures. The "manifestations" detected by this comparison, such as an abnormally high rate of programmer hours per line of source code, then serve as input to each expert system. Finally, each system attempts to determine the reasons, such as low productivity, for any abnormal software development patterns. These systems can be updated as the environment changes and as more is learned in the field of software engineering.

The two systems, which solve the same problem, were built using different methods: rule-based deduction and frame-based abduction. A comparison was performed to determine which method better suits the needs of this field. It was found that both systems performed moderately well, but the rule-based deduction system using simple rules provided more complete solutions than did the frame-based abduction system. JAO

8978

SECTION 4 - MODELS AND MEASURES

8978

4.1 "A CHILD'S GARDEN OF COMPLEXITY MEASURES," S. F. LANGE
    (PAPER PREPARED FOR THE UNIVERSITY OF MARYLAND,
    DECEMBER 1978), 31 PAGES

This paper investigates several forms of the McCabe measure
of program complexity. The McCabe measure is defined, its
relationship to structured programming is examined, and the
McCabe concept itself is evaluated by comparison with data
obtained from actual FORTRAN programs. Programmer-rated
difficulty is found to be highly correlated with all forms
of the McCabe complexity measure considered in this analy-
sis. Furthermore, these results support the proposed rela-
tionship between structured programming and the various
forms of the McCabe complexity measure.

4.2 "A META-MODEL FOR SOFTWARE DEVELOPMENT RESOURCE
    EXPENDITURES," J. W. BAILEY AND V. R. BASILI,
    PROCEEDINGS OF THE FIFTH INTERNATIONAL CONFERENCE ON
    SOFTWARE ENGINEERING. NEW YORK: COMPUTER SOCIETIES
    PRESS, 1981, 10 PAGES

This technical paper describes an effort to produce a model
of software development resource expenditures that can be
generalized to a number of situations. Many models have
been proposed over the last several years. However, experi-
ence has shown that differences in the data collected, types
of projects developed, and environmental factors limit the
transportability of these models from one organization to
another. This conclusion is reasonable because a model de-
veloped in any given environment will reflect only the im-
pact of factors that have a variable effect in that
environment. Factors that are constant in that environment
(and therefore do not affect productivity) may have differ-
ent or variable effects in another environment.

This paper describes a model-generation process that permits
the development of a resource estimation model for any par-
ticular organization. The process provides the capability
to produce a model that is tailored to the organization and
can be expected to be more effective than any model orig-
inally developed for another environment. The model is
demonstrated here using data collected by the SEL at GSFC.
JAO

This technical paper also appears in SEL-82-004, Collected
Software Engineering Papers: Volume I, July 1982.

8978

4.3 "A MODEL OF THE SOFTWARE LIFE CYCLE," K. FREBURGER
(PAPER PREPARED FOR THE UNIVERSITY OF MARYLAND,
DECEMBER 1978), 27 PAGES

This paper describes the software life cycle and its relationship to the Rayleigh curve model of resource expenditures. The software life cycle is defined as a sequence of phases or steps through which a software system progresses until development is completed. Resource expenditures during the software life cycle can be modeled with a Rayleigh curve. This model allows staffing levels, development time, and total cost of development to be estimated.

4.4 A STUDY OF THE MUSA RELIABILITY MODEL, SEL-80-005, A. M. MILLER, NOVEMBER 1980, 94 PAGES

This document describes a study in which the Musa reliability model was applied to three software projects developed for GSFC, with the goal of determining whether the model could be used in the flight dynamics environment as a software management tool. One purpose of the model is to predict the total number of errors in a piece of software undergoing testing. Actual times between failures and their associated run times were fitted to the Musa equation in an iterative procedure. Of the three projects studied, the results for one converged to a value 25 percent higher than the actual number of errors; the other two did not converge at all.

The document discusses the assumptions underlying the model and evaluates the characteristics of the environment that could affect these assumptions. Suggestions are offered about changes that could be made in the environment to better meet the assumptions. This document was originally prepared as a Master's Thesis at the University of Maryland. NSTF

8978

4.5 "A SURVEY OF SEVERAL RELIABILITY MODELS," A. M. MILLER (PAPER PREPARED FOR THE UNIVERSITY OF MARYLAND, DECEMBER 1978), 19 PAGES

This paper describes and compares three prominent software reliability models. It outlines the assumptions, limitations, and mathematical formulations of these models. The models studied are those developed by the following researchers:

- M. L. Shooman
- J. D. Musa
- B. Littlewood

The predictions of the models are compared with the results obtained from an actual software project studied by the SEL. Firm conclusions cannot be drawn because of the inadequacy of the data. However, recommendations are made for changes in the data collection process to facilitate this type of analysis.

4.6 AN APPRAISAL OF SELECTED COST/RESOURCE ESTIMATION MODELS FOR SOFTWARE SYSTEMS, SEL-80-007, J. F. COOK AND F. E. McGARRY, DECEMBER 1980, 41 PAGES

This document presents the results of an evaluation and comparison of seven cost/resource estimation models based on SEL data. The following models were considered:

- Doty
- Walston/Felix
- Tecolote
- GRC
- SLIM
- PRICE S3
- SEL Meta-Model

The validity of the theoretical bases of these models was not analyzed. The objective of the appraisal was simply to determine how well SEL data conformed to the predictions of various models. This document was also issued as Goddard Space Flight Center document X-582-81-1. NTIS

4.7 "ANALYZING MEDIUM SCALE SOFTWARE DEVELOPMENT,"
V. R. BASILI AND M. V. ZELKOWITZ, PROCEEDINGS OF THE
THIRD INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING.
NEW YORK: COMPUTER SOCIETIES PRESS, MAY 1978, 8 PAGES

This technical paper surveys SEL research activities in
software engineering. The collection and analysis of data
from software development projects is necessary for the de-
finitive evaluation of software engineering methodologies
and techniques. This paper describes the structure of the
SEL and some of the early projects that were monitored. It
also discusses the application of this data to resource
utilization models and reliability studies. The principal
contribution of the SEL, as reported in the paper, is the
establishment of a facility for collecting the detailed data
necessary for these analyses. JAO

4.8 APPLICABILITY OF THE RAYLEIGH CURVE TO THE SEL ENVIRONMENT, SEL-78-007, T. E. MAPP, DECEMBER 1978, 27 PAGES

This document reviews the resource utilization model for software development, which is based on the Rayleigh curve developed by Norden and Putnam. A Rayleigh curve is fit to data provided by the SEL. Parabolas, trapezoids, and straight lines are also fit to the same data. The parabola and trapezoid give about as good a fit as the Rayleigh curve. Therefore, this document concludes that while the Rayleigh curve may be an appropriate model for resource expenditures, it is not necessarily the best model for small- to medium-size projects. NSTF

4.9 "CAN THE PARR CURVE HELP WITH MANPOWER DISTRIBUTION AND
    RESOURCE ESTIMATION PROBLEMS?", V. R. BASILI AND
    J. BEANE, JOURNAL OF SYSTEMS AND SOFTWARE,
    FEBRUARY 1981, VOL. 2, NO. 1, 11 PAGES

This technical paper analyzes the resource utilization model
developed by Parr. The curve predicted by the model is com-
pared with several other curves, including the Rayleigh
curve, a parabola, and a trapezoid, with respect to how well
they fit manpower utilization. The evaluation is performed
for several flight dynamics projects of the 6- to 12-man-year
effort range that were studied by the SEL.

The conclusion drawn is that the Parr curve can be made to
fit the data better than the other curves. However, because
of the noise in the data, it is difficult to confirm the
shape of the manpower distribution from the data alone and
therefore difficult to validate any particular model. More-
over, since the parameters used in the curve are not easily
calculable or estimable from known data, the curve is not
effective for resource estimation. JAO

This technical paper also appears in SEL-82-004, Collected
Software Engineering Papers: Volume I, July 1982.

8978

4.10 "COMPARISON OF REGRESSION MODELING TECHNIQUES FOR
      RESOURCE ESTIMATION," D. N. CARD, COMPUTER SCIENCES
      CORPORATION TECHNICAL MEMORANDUM, NOVEMBER 1982,
      21 PAGES

This technical memorandum presents the results of a study
conducted to compare three alternative regression procedures
by examining the results of their application to one com-
monly accepted equation for resource estimation. Linear,
Log-Linear, and Nonlinear procedures were considered. The
memorandum summarizes the data studies, describes the re-
source estimation equation, explains the regression proce-
dures, and compares the results obtained from the
procedures. The regression procedures were evaluated with
respect to numerical accuracy, conceptual accuracy, and
computational cost. This study is based on data collected
from 22 flight dynamics software projects studied by the
SEL. FEM

_____

This technical paper also appears in SEL-83-003, Collected
Software Engineering Papers: Volume II, November 1983.

4-11

8978

4.11 "DESIGNING A SOFTWARE MEASUREMENT EXPERIMENT,"
V. R. BASILI AND M. V. ZELKOWITZ, PROCEEDINGS OF THE
SOFTWARE LIFE CYCLE MANAGEMENT WORKSHOP,
SEPTEMBER 1977, 13 PAGES

This technical paper explains the research approach employed
by the SEL to study the development of actual software de-
velopment projects. The following types of experiments are
performed by the SEL:

- Screening
- Semicontrolled
- Controlled

This paper discusses these experimental designs, potential
confounding effects, and the statistical techniques used to
evaluate results. The effects on software developers of
both learning during the experiment and an awareness of the
experimental process itself are examined in detail. Fully
controlled experiments are especially difficult to implement
in a production environment, but sufficient control is pos-
sible to evaluate the effects of software development meth-
odologies. JAO

4.12 "EARLY ESTIMATION OF RESOURCES EXPENDITURES AND
       PROGRAM SIZE," D. N. CARD, COMPUTER SCIENCES CORPORA-
       TION, TECHNICAL MEMORANDUM, JUNE 1982, 24 PAGES

This technical memorandum evaluates the suitability of sev-
eral software measures as estimators of resource expendi-
tures and program size early in the software life cycle.
The estimating equation based on the most commonly employed
measure, lines of source code, is explained and its limita-
tions are identified. Several alternative measures are in-
vestigated and found to give good results. The memorandum
also includes computer-generated output of the least-squares
regression analyses upon which the conclusions are based.
FEM

_____

This technical paper also appears in SEL-83-003, Collected
Software Engineering Papers: Volume II, November 1983.

8978

4.13   "ERROR AND CHANGE ANALYSIS," D. M. WEISS, NAVAL
       RESEARCH LABORATORY, TECHNICAL MEMORANDUM,
       JULY 1979, 56 PAGES

This technical memorandum discusses the essential concepts
upon which any analysis of error and change data must be
based.   These concepts are the motivation for error/change
analyses, the generation of hypotheses, and data collection
procedures.   The measures and distributions relevant to
these analyses are defined and ranked according to their
importance and the difficulty with which they are obtained.
The memorandum also explains the derivations of these meas-
ures and provides the outlines of software functions for
calculating them.   In short, this memorandum provides useful
information for planning an analysis of error/change data.

4.14 "EVALUATING AND COMPARING SOFTWARE METRICS IN THE
      SOFTWARE ENGINEERING LABORATORY," V. R. BASILI AND
      T. PHILLIPS, PROCEEDINGS OF THE ACM SIGMETRICS
      SYMPOSIUM/WORKSHOP: QUALITY METRICS, MARCH 1981,
      19 PAGES

This technical paper describes an effort to identify the
best measures of software development effort and software
complexity. Four software projects studied by the SEL pro-
vide the data for the analysis. The data is screened to
ensure its validity. Next, estimating equations are derived
for effort and errors using the various measures studied in
the analysis. Correlations are shown to increase as the
reliability of the data increases due to screening. Thus, a
procedure is demonstrated for removing noise from the data
and making possible meaningful comparisons of software
metrics. JAO

This technical paper also appears in SEL-82-004, Collected
Software Engineering Papers: Volume I, July 1982.

8978

4.15 "EVALUATING AUTOMATABLE MEASURES FOR SOFTWARE DEVELOP-
MENT," V. R. BASILI AND R. REITER, PROCEEDINGS OF THE
WORKSHOP ON QUANTITATIVE SOFTWARE MODELS FOR RELI-
ABILITY, COMPLEXITY, AND COST. NEW YORK: COMPUTER
SOCIETIES PRESS, 1979, 10 PAGES

This technical paper describes an approach to developing and evaluating automatable software measures. The experience of the SEL has shown that software data collection is an expensive activity that can significantly affect the software development process. Costs and effects can be minimized and data quality can be improved by automating the collection of measures wherever possible.

This paper presents a set of automatable measures that were implemented and evaluated in a controlled experiment. The measures include computer job steps, program changes, program size, and software complexity. The results of the experiment indicate that the automated collection of these measures can be implemented effectively in production environments. JAO

4-16

8978

4.16 EVALUATION OF MANAGEMENT MEASURES OF SOFTWARE DEVELOPMENT, SEL-82-001, G. PAGE, D. N. CARD, AND F. E. McGARRY, SEPTEMBER 1982, VOL. 1: 143 PAGES, VOL. 2: 379 PAGES

This two-volume document reports the results of an evaluation of a large set of software development measures relevant to the GSFC environment. The purposes of the analysis were to characterize the current software development process in one environment by identifying important qualities and corresponding measures and to evaluate the effectiveness of specific tools and techniques in this environment. The measures studied were counts, ratios, and management-supplied ratings of various elements of the software development process. The measures are high level in that each describes some aspect of an entire software project (or a large part of it) rather than individual components of the project.

Volume 1 explains a conceptual model of software development, the data classification scheme, and the analytic procedures. Factor analysis, cluster analysis, and a test of normality were used. This volume summarizes the results of those analyses and recommends specific software measures for collection and monitoring. Volume 1 also reproduces in full the results of the computer analyses.

Volume 2 presents a detailed description of the data analyzed, including definitions of measures, lists of values, and summary statistics. Although the information contained in Volume 2 was essential to the development of the explanation and summary presented in Volume 1, it is not essential to the understanding of that explanation and summary. However, Volume 2 is useful in its own right as a source of data and a reference for future research. This document was also issued as Computer Sciences Corporation document CSC/TM-82/6063. FEM

8978

4.17 AN APPROACH TO SOFTWARE COST ESTIMATION, SEL-83-001,
F. E. MCGARRY, G. PAGE, D. N. CARD, ET AL.,
FEBRUARY 1984, 73 PAGES

This document describes the general procedures for software cost estimation in any environment. First, the basic concepts of work and effort estimation are explained, some popular resource estimation models are reviewed, and the accuracy of resource estimates is investigated. Next, general guidelines are presented for cost estimation throughout the software life cycle. The sources of information and relevant parameters available during each phase cycle are identified. Finally, a comprehensive software cost prediction procedure based on the experiences of the SEL in the flight dynamics area and incorporating management expertise, cost models, and historical data is provided. The methodology developed incorporates these elements into a customized management tool for software cost prediction. This document was also issued as Computer Sciences Corporation document CSC/TM-83/6076. NTIS

4.18 MANAGER'S HANDBOOK FOR SOFTWARE DEVELOPMENT, SEL-84-001, W. W. AGRESTI, F. E. MCGARRY, D. N. CARD, ET AL., APRIL 1984, 59 PAGES

This document presents methods and aids for the management of software development projects. The recommendations are based on analyses and experiences of the SEL with flight dynamics software development. The management aspects of the following subjects are described:

- Organizing the project
- Producing a development plan
- Estimating costs
- Scheduling
- Staffing
- Preparing deliverable documents
- Using management tools
- Monitoring the project
- Conducting reviews
- Auditing
- Testing
- Certifying

This document was also issued as Computer Sciences Document CSC/TM-83/6177. NTIS

4-19

8978

4.19 MEASURES AND METRICS FOR SOFTWARE DEVELOPMENT, SEL-83-002, D. N. CARD, F. E. McGARRY, G. PAGE, ET AL., MARCH 1984, 80 PAGES

This document reports the evaluations of and recommendations for the use of software development measures based on the practical and analytical experience of the SEL. It describes the basic concepts of measurement and a system of classification for measures. The principal classes of measures defined are explicit, analytic, and subjective. Some of the major software measurement schemes appearing in the literature are reviewed. The applications of specific measures in a production environment are explained. These applications include the following:

- Prediction and Planning
- Review and Assessment
- Evaluation and Selection

An appendix describes the use of software development histories to manage ongoing software development projects. This document was also issued as Computer Sciences Corporation document CSC/TM-83/6061. NTIS

4.20 "MEASURING SOFTWARE DEVELOPMENT CHARACTERISTICS IN THE LOCAL ENVIRONMENT," V. R. BASILI AND M. V. ZELKOWITZ, COMPUTERS AND STRUCTURES, AUGUST 1978, VOL. 10, 5 PAGES

This technical paper discusses the role of data collection in forecasting and monitoring software development projects in a production environment. The specific procedures of the SEL are reviewed, and SEL data collection forms are described. The paper also gives some examples of analyses that can be performed to support managing, understanding, and characterizing software development. The sample analyses identify specific measures for collection. JAO

This technical paper also appears in SEL-82-004, Collected Software Engineering Papers: Volume I, July 1982.

4-21

4.21 "METRIC ANALYSIS AND DATA VALIDATION ACROSS FORTRAN
PROJECTS," V. R. BASILI, R. W. SELBY, AND T. PHILLIPS,
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, NOVEMBER
1983

This technical paper reports the results of an analysis of
the relationship of Halstead measures, McCabe complexity
measures, and other software measures to software develop-
ment effort and errors. Effort is defined in terms of
staff-hours from the establishing of functional specifica-
tions through acceptance testing. Errors are counted dis-
cretely and weighed according to effort to correct. The
data studied was collected by the SEL in a production en-
vironment. Cross-checks of the data indicated a need for
large-scale data validation. The strongest correlations
were obtained when the modules of individual programmers
were considered independently. However, neither Halstead's
effort measure, McCabe's cyclomatic complexity measure, nor
lines of source code was convincingly more accurate as an
estimator than the others. This document was also issued as
University of Maryland Technical Report TR-1228. JAO

8978

4.22 MODELS AND METRICS FOR SOFTWARE MANAGEMENT AND ENGINEERING, SEL-80-008, V. R. BASILI, 1980, 349 PAGES

This document is a tutorial on quantitative methods of software management and engineering. A quantitative methodology is needed to evaluate, control, and predict software development and maintenance costs. This quantitative approach allows cost, time, and quality tradeoffs to be made in a systematic manner. The tutorial focuses on numerical product-oriented measures such as size, complexity, and reliability and on resource-oriented measures such as cost, schedules, and resources. Twenty articles from software engineering literature are reprinted in this document. The articles are organized into the following sections:

- Resource models
- Changes and errors
- Product metrics
- Data collection

Successful application of these techniques, however, requires a thorough knowledge of the project under development and any assumptions made. Only then can these techniques augment good managerial and engineering judgment. JAO

This document was published as the IEEE tutorial, Models and Metrics for Software Management and Engineering, New York: Computer Societies Press, 1980.

8978

4.23 "MODELS AND METRICS FOR SOFTWARE MANAGEMENT AND
ENGINEERING," V. R. BASILI, ASME ADVANCES IN COMPUTER
TECHNOLOGY, JANUARY 1980, VOL. 1, 12 PAGES

This technical paper attempts to characterize several quan-
titative models and measures of the software development
process. These models and measures deal with various as-
pects of the software process and product, including re-
source estimation, complexity, reliability, and size. The
relationship of these models and measures to the software
development life cycle is also discussed. Finally, the ex-
tent to which the various models have been applied in pro-
duction environments and the success they have achieved is
indicated. JAO

This technical paper also appears in SEL-82-004, Collected
Software Engineering Papers: Volume I, July 1982.

4.24 "PROGRAMMING MEASUREMENT AND ESTIMATION IN THE SOFTWARE ENGINEERING LABORATORY," V. R. BASILI AND K. FREBURGER, JOURNAL OF SYSTEMS AND SOFTWARE, FEBRUARY 1981, VOL. 2, NO. I, II PAGES

This technical paper presents an examination of a set of basic relationships among various software development measures, including size, effort, project duration, staff size, and productivity. Correlations among these measures are computed. The data used comes from 15 flight dynamics software development projects studied by the SEL. Certain relationships are derived in the form of equations, and these equations are compared with a set derived by Walston and Felix for IBM Federal Systems Division project data. Logarithmic transformations were performed on the data for some analyses. Although the equations do not have the same coefficients, they are seen to have similar exponents. In fact, the SEL-derived equations tend to be within one standard error of the estimates provided in the IBM equations. JAO

This technical paper also appears in SEL-82-004, Collected Software Engineering Papers: Volume I, July 1982.

8978

4.25   "RESOURCE ESTIMATION FOR MEDIUM SCALE SOFTWARE PROJ-
       ECTS," M. V. ZELKOWITZ, PROCEEDINGS OF THE TWELFTH CON-
       FERENCE ON THE INTERFACE OF STATISTICS AND COMPUTER
       SCIENCE.   NEW YORK: COMPUTER SOCIETIES PRESS, 1979,
       6 PAGES

This technical paper describes the analysis of resource es-
timation techniques that is being performed by the SEL.   The
data used in the analysis is collected from medium-scale
flight dynamics software development projects at GSFC.   A
procedure to forecast accurately the cost and development
time of these projects would be a valuable management tool
in this environment.   This paper documents a specific at-
tempt to verify the resource estimation model based on the
Rayleigh curve that was developed by Norden and Putnam.   JAO

This technical paper also appears in SEL-82-004, Collected
Software Engineering Papers: Volume I, July 1982.

8978

4.26 "RESOURCE MODEL TESTING AND INFORMATION,"
I. M. WILLIAMSON, NAVAL RESEARCH LABORATORY, TECHNICAL
MEMORANDUM, JULY 1979, 41 PAGES

This technical memorandum reviews seven published resource
estimation models. It identifies the parameters, explains
the algorithm, and gives an example of application for each
model. The models studied are

- NARDAC
- Doty
- Tecolote
- Aron
- Walston/Felix
- GRC
- Wolverton

Limitations and problems encountered in the application ex-
amples or published presentations of these models are dis-
cussed.

4.27 "SOFTWARE ENGINEERING LABORATORY RELATIONSHIPS FOR
     PROGRAMMING MEASUREMENT AND ESTIMATION," V. R. BASILI,
     UNIVERSITY OF MARYLAND, TECHNICAL MEMORANDUM,
     OCTOBER 1979, 24 PAGES

This technical memorandum reproduces a series of graphs
showing the relationships among several software measures.
The graphs were produced by regression analyses of SEL data.
The relationships illustrated are those investigated by
Walston and Felix. See also Section 4.23.

8978

4.28 "SOFTWARE ERRORS AND COMPLEXITY: AN EMPIRICAL INVESTI-GATION," V. R. BASILI AND B. T. PERRICONE, COMMUNICA-TIONS OF THE ACM, JANUARY 1984, 11 PAGES

This paper reports the results of an analysis of error data obtained from a flight dynamics software project studied by the SEL. The distributions of errors by type and location are identified and discussed. Correlations among module size, complexity, and error rate are then described and evaluated. Modified and new modules are shown to have simi-lar error characteristics. An alternative error classifica-tion scheme is developed. Finally, an attempt is made to compare these results with those of other researchers in the field. FEM

This technical paper also appears in SEL-83-003, Collected Software Engineering Papers: Volume II, November 1983, and was also issued as University of Maryland Technical Report, TR-1195, August 1982.

8978

4.29 "SOME TESTS OF HALSTEAD MEASURES," G. HISLOP
(PAPER PREPARED FOR THE UNIVERSITY OF MARYLAND,
DECEMBER 1978), 26 PAGES

This paper attempts to evaluate the "software science" measures devised by Halstead. It first explains the concepts of operators and operands and then defines some complex measures composed of operators and operands. The predicted values of these measures are correlated with actual values for sample programs. The results of this analysis indicate that Halstead measures are relevant to some aspects of software development but not to all.

4.30 "THE NATURE, ORGANIZATION, MEASUREMENT, AND MANAGEMENT OF SOFTWARE COMPLEXITY," R. W. REITER (PAPER PREPARED FOR THE UNIVERSITY OF MARYLAND, DECEMBER 1976), 47 PAGES

This paper is a discussion of the relationship of software complexity to software development and maintenance. The sources, or factors, of software complexity are also considered as well as complexity measures. The complexity of a software system is closely related to the difficulty of understanding that system and the way in which its elements fit and function together. The term "complexity," as defined in this paper, does not specifically refer to the difficulty of algorithms.

4-31

4.31 THE RAYLEIGH CURVE AS A MODEL FOR EFFORT DISTRIBUTION OVER THE LIFE OF MEDIUM SCALE SOFTWARE SYSTEMS, SEL-81-012, G. O. PICASSO, DECEMBER 1981, 153 PAGES

This document discusses some of the factors affecting the accuracy of resource models applied to medium-scale software systems. Putnam has shown that the Rayleigh curve is an adequate model for the life-cycle effort distribution of large-scale systems. Previous investigations of the applicability of this model to medium-scale software development efforts have met with mixed results. The results of the earlier investigations are confirmed in this analysis. The reasons for the failure of the models are found in the sub-cycle (phase) effort data. There are four contributing factors: uniqueness of the environment studied, the influence of holidays, varying management techniques, and differences in the data studied. This document was also issued as University of Maryland Technical Report TR-1186. FEM

8978

4.32 THE SOFTWARE ENGINEERING LABORATORY: RELATIONSHIP EQUATIONS, SEL-79-002, K. FREBURGER AND V. R. BASILI, MAY 1979, 67 PAGES

This document presents the results of an analysis of several factors affecting software development. The analysis was based on data collected by the SEL. Relationships among the following measures were studied:

- Total effort (staff-months)
- Lines of delivered code (thousands)
- Lines of developed code (thousands)
- Percentage of developed code
- Number of modules
- Number of developed modules
- Percentage of developed modules
- Project duration (months)
- Pages of documentation
- Productivity
- Average staff size

Estimating equations were derived from the measures by statistical analysis and were then compared with results obtained by Walston and Felix in a similar study. This document was also issued as University of Maryland Technical Report TR-764. NSTF

4.33 MONITORING SOFTWARE DEVELOPMENT THROUGH DYNAMIC VARIABLES, SEL-83-006, C. W. DOERFLINGER, NOVEMBER 1983, 110 PAGES

This document describes research conducted by the SEL on the use of dynamic variables as a tool for monitoring software development. The intent of the project, which examined several FORTRAN projects with similar profiles, was to identify project-independent measures. The projects developed serve similar functions, and because the projects are similar, some underlying relationships exist that are invariant between the projects. These relationships, once well defined, may be used to compare the development of different projects to determine whether they are evolving in the same way previous projects in this environment evolved. This document was originally prepared as a Master's Thesis at the University of Maryland. FEM

4.34 "MONITORING SOFTWARE DEVELOPMENT THROUGH DYNAMIC VARIABLES," C. W. DOERFLINGER AND V. R. BASILI, PROCEEDINGS OF THE SEVENTH INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, NOVEMBER 1983, 27 PAGES

This paper summarizes the SEL document (SEL-83-006) of the same name. It describes research conducted by the SEL on the use of dynamic variables as a tool for monitoring software development. The intent of the project, which examined several FORTRAN projects with similar profiles, was to identify project-independent measures. The projects developed serve similar functions, and because the projects are similar, some underlying relationships exist that are invariant between the projects. These relationships, once well defined, may be used to compare the development of different projects to determine whether they are evolving in the same way previous projects in this environment evolved. JAO

This technical paper also appears in SEL-83-003, Collected Software Engineering Papers: Volume II, November 1983.

4-35

8978

4.35 "MEASURING SOFTWARE TECHNOLOGY," W. W. AGRESTI,
F. E. MCGARRY, D. N. CARD, ET AL., PROGRAM TRANS-
FORMATION AND PROGRAMMING ENVIRONMENTS. NEW YORK:
SPRINGER-VERLAG, 1984, 6 PAGES

This paper summarizes the results of several recent SEL re-
search efforts. The areas of software engineering discussed
are programmer productivity, cost models, and technology
evaluations. This paper stresses the importance of estab-
lishing an organizational memory to provide a reference for
evaluating software engineering techniques. The SEL data
collection program is outlined as an example of such a mech-
anism. JAO

This technical paper also appears in SEL-83-003, Collected
Software Engineering Papers: Volume II, November 1983.

8978

4.36 DEFINITION OF SPECIFICATION MEASURES FOR THE SOFTWARE ENGINEERING LABORATORY (SEL), CSC/TM-84/6085, W. W. AGRESTI, COMPUTER SCIENCES CORPORATION, JUNE 1984, 49 PAGES

This report identifies requirements specification measures for potential application in the SEL. Eighty-seven candidate measures are defined: 29 derived from documentation that is routinely produced during the flight dynamics software development process and 58 based on information that is available but is not now regularly captured in documentation. The latter set of measures can be obtained when the requirements are recast in a new representation, the Composite Specification Model, which is introduced in this document. Of the 87 measures, 16 are recommended for use. FEM

4.37 "CHARACTERISTICS OF FORTRAN MODULES," D. N. CARD,
Q. L. JORDAN, AND V. E. CHURCH, COMPUTER SCIENCES
CORPORATION, TECHNICAL MEMORANDUM, JUNE 1984, 62 PAGES

This study analyzes the characteristics of a large sample of
FORTRAN modules produced by professional programmers. The
proper organization and content of a software module are
basic concerns of software developers. Although many stra-
tegies and standards for software development are in use,
few are based on any empirical evidence. The data studied
was collected by the SEL. It includes error reports, staff
charges, and source code measures. This study attempts to
determine whether or not structural and quality differences
exist among different classes of software and how knowledge
of structural characteristics can be used to maximize soft-
ware quality.

8978

4.38 STRUCTURAL COVERAGE OF FUNCTIONAL TESTING, TR-1442, V. R. BASILI AND J. RAMSEY, UNIVERSITY OF MARYLAND, TECHNICAL REPORT, SEPTEMBER 1984

This technical report describes a study directed to understanding and improving the acceptance test process in the NASA/GSFC SEL environment. A large, commercially developed FORTRAN program was modified to produce structural coverage metrics. The modified program was executed on a set of functionally generated acceptance tests and a large sample of operational usage cases. The resulting structural coverage metrics are combined with fault and error data to evaluate structural coverage in the SEL environment.

It is shown that, in this environment, the functionally generated tests seem to be a good approximation of operational use. The relative proportions of the exercised statement subclasses (executable, assignment, CALL, DO, IF, READ, WRITE) change as the structural coverage of the program increases. A method is proposed for determining whether two sets of input data exercise a program in a similar manner.

Evidence is also provided implying that, in this environment, faults revealed in a procedure are independent of the number of times the procedure is executed and that it may be reasonable to use procedure coverage in software models that use statement coverage. Finally, the evidence suggests that it may be possible to use structural coverage to aid the management of the acceptance test process. FEM

4.39 "EVALUATING SOFTWARE DEVELOPMENT BY ANALYSIS OF
CHANGES: SOME DATA FROM THE SOFTWARE ENGINEERING
LABORATORY," D. M. WEISS AND V. R. BASILI, <u>IEEE
TRANSACTIONS ON SOFTWARE ENGINEERING</u>, FEBRUARY 1985,
12 PAGES

This paper describes the application of an effective data
collection methodology for evaluating software development
methodologies to five different software development proj-
ects. Results and data from three of the projects are pre-
sented. Goals of the data collection included character-
izing changes, errors, projects, and programmers; identi-
fying effective error detection and correction techniques;
and investigating ripple effects.

The data collected consisted of changes (including error
corrections) made to the software after code was written and
baselined, but before testing began. Data collection and
validation were concurrent with software development.
Changes reported were verified by interviews with program-
mers. Analysis of the data showed patterns that were used
in satisfying the goals of the data collection. JAO

8978

4.40 "CRITERIA FOR SOFTWARE MODULARIZATION," D. N. CARD,
       G. PAGE, AND F. E. MCGARRY, PROCEEDINGS OF THE EIGHTH
       INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING,
       AUGUST 1985, 6 PAGES

This paper reports an attempt to determine the effectiveness
of two widely used criteria for software modularization,
strength and size, in reducing fault rate and development
cost. The study was prompted by a central issue in prog-
ramming practice that involves determining the appropriate
size and information content of a software module. Data
from 453 FORTRAN modules developed by professional program-
mers were analyzed. The results indicated that module
strength is a good criterion with respect to fault rate,
whereas arbitrary module size limitations inhibit programmer
productivity. This analysis is a first step toward defining
empirically based standards for software modularization. JAO

8978

4.41 "CALCULATION AND USE OF AN ENVIRONMENT'S CHARACTER-
       ISTIC SOFTWARE METRIC SET," V. R. BASILI AND
       R. W. SELBY, JR., PROCEEDINGS OF THE EIGHTH
       INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING,
       AUGUST 1985, 6 PAGES

This paper presents an approach for customizing a character-
istic set of software metrics to an environment, since both
cost/quality goals and production environments differ. The
approach is applied in the SEL to 49 candidate process and
product metrics of 652 modules from six projects (of 51,000
to 112,000 lines). For this particular environment, the
method yielded the characteristic metric set (source lines,
fault correction effort per executable statement, design
effort, code effort, number of I/O parameters, number of
versions). The uses examined for a characteristic metric
set include forecasting the effort for development, modifi-
cation, and fault correction of modules based on historical
data. JAO

8978

4.42 "FINDING RELATIONSHIPS BETWEEN EFFORT AND OTHER
VARIBLES IN THE SEL," V. R. BASILI AND
N. M. PANLILIO-YAP, PROCEEDINGS OF THE INTERNATIONAL
COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE,
OCTOBER 1985, 7 PAGES

This study examines the relationship between effort and
other variables for 23 SEL projects that were were developed
for NASA/GSFC. These variables fell into two categories:
those that can be determined in the early stages of project
development and may therefore be useful in a baseline equa-
tion for predicting effort in future projects, and those
that can be used mainly to characterize or evaluate effort
requirements and thus enhance our understanding of the soft-
ware development process in this environment. Some results
of the analyses are presented in this paper. JAO

8978

4.43 INVESTIGATION OF SPECIFICATION MEASURES FOR THE SOFTWARE ENGINEERING LABORATORY, SEL-84-003, W. W. AGRESTI, V. E. CHURCH, AND F. E. MCGARRY, DECEMBER 1984

This document presents an investigation of requirements specification measures for potential application in the SEL. Eighty-seven candidate measures are defined; sixteen are recommended for use. Most measures are derived from a new representation, the Composite Specification Model, which is introduced in this document. The results of extracting the specification measures from the requirements of a real system are described. NTIS

4-44

SECTION 5 - TECHNOLOGY EVALUATIONS

5-1

5.1 A DEMONSTRATION OF AXES FOR NAVPAK, SEL-77-005, M. HAMILTON AND S. ZELDIN, SEPTEMBER 1977, 84 PAGES

This document describes the results of a demonstration project employing a system specification language tool to respond to a flight dynamics software development requirement. The tool, AXES, was developed by Higher Order Software, Inc. Specific areas were selected for the demonstration from the NAVPAK software requirements. The complexity of these requirements and the effort required to obtain a full understanding of them prevented the development of a complete design within the demonstration period. This document describes the following aspects of the demonstration project:

- Principles of HOS/AXES
- Description of NAVPAK
- HOS/AXES representation of NAVPAK specifications

The appendixes contain some unrelated examples and other background material. This document was also issued as Higher Order Software, Inc., Technical Report 9.

8978

5.2 EVALUATING SOFTWARE DEVELOPMENT BY ANALYSIS OF CHANGE DATA, SEL-81-011, D. M. WEISS, NOVEMBER 1981, 272 PAGES

This document reports the results of an analysis of change data from five different software development projects in two different environments. A common data collection methodology was applied at both GSFC and the Naval Research Laboratory (NRL). This document describes the data collection methodology employed, software projects studied, and the effects of changes on software development.

The results of this study indicate that the data collection methodology is effective and easily extendable to new software development environments. Although the GSFC and NRL environments differed somewhat in their objectives and approach to software development, the software produced by both groups was similar with respect to changes and errors. The results presented in this document include: (1) distributions of causes of change, sources of errors, and difficulty of finding errors and (2) tabulations of changes according to number of components changed, changes according to subsystem, difficulty of change (error) according to source of change (error), and source of error according to programmer. This document was also issued as University of Maryland Technical Report TR-1120. FEM

5.3 EVALUATION OF AN INDEPENDENT VERIFICATION AND VALIDATION (IV&V) METHODOLOGY FOR FLIGHT DYNAMICS, SEL-81-110, G. PAGE, F. E. McGARRY, AND D. N. CARD, JUNE 1985, 53 PAGES

This document describes an experiment in the application of an independent verification and validation (IV&V) methodology to the development of flight dynamics software at GSFC. IV&V is the systematic evaluation of computer software by an organization that is independent of the development organization. IV&V is expected to provide earlier error detection and better quality control over the development process.

This document describes the environment, staffing, and results of the experiment. Costs and error rates are compared with those of similar projects developed without IV&V. An IV&V methodology is found to be appropriate for very large projects and for those with high reliability requirements. This document was also issued as Computer Sciences Corporation document CSC/TM-85/6045. FEM

The previous version of this document was Performance and Evaluation of an Independent Software Verification and Integration Process, SEL-81-010, G. Page and F. E. McGarry, May 1981.

8978

5.4 EVALUATION OF DRAPER NAVPAK SOFTWARE DESIGN, SEL-78-003, K. TASAKI AND F. E. MCGARRY, JUNE 1978, 23 PAGES

This document describes the results of an evaluation of a software design demonstration project. The design methodology was developed by the Charles Stark Draper Laboratories (CSDL). A sample flight dynamics project was designed by CSDL from the NAVPAK software requirements using this methodology. The end product was evaluated to determine the effectiveness of the CSDL methodology. <u>FEM</u>

5.5 EVALUATION OF THE CAINE, FARBER, AND GORDON PROGRAM
DESIGN LANGUAGE (PDL) IN THE GODDARD SPACE FLIGHT
CENTER (GSFC) CODE 580 SOFTWARE DESIGN ENVIRONMENT,
SEL-79-004, C. E. GOOREVICH, A. L. GREEN, AND
W. J. DECKER, SEPTEMBER 1979, 46 PAGES

This document reports the results of a study of the useful-
ness of program design languages (PDLs) for flight dynamics
software development at GSFC. The following PDLs were ex-
amined and compared:

- Telemetry Computation Branch PDL

- Linger and Mills PDL

- Caine, Farber, and Gordon PDL

The last PDL was selected for intensive study. Its advan-
tages and disadvantages in the flight dynamics environment
were evaluated. Appendixes include examples of the use of
the Caine, Farber, and Gordon PDL and the processor output.
This document was also issued as Computer Sciences Corpora-
tion document CSC/TM-79/6263. NSTF

8978

5.6 "GSFC NAVPAK DESIGN HIGHER ORDER LANGUAGES STUDY: ADDENDUM," P. A. SCHEFFER AND C. E. VELEZ, MARTIN MARIETTA CORPORATION, TECHNICAL MEMORANDUM, SEPTEMBER 1977, 170 PAGES

This technical memorandum reproduces the computer output on which the software design languages evaluation described in Section 2.5 was based in part. The materials reproduced include the Problem Statement Language (PSL) input (or source) and the following types of output produced by the PSL processor:

- Formatted problem statement
- Cross-reference index
- Data-process report
- Consists matrix
- Extended picture report

These tables and reports can be used to identify errors and ambiguities in the PSL representation of the software design.
FEM

5-7

5.7 GSFC NAVPAK DESIGN SPECIFICATION LANGUAGES STUDY,
     SEL-77-004, P. A. SCHEFFER AND C. E. VELEZ,
     OCTOBER 1977, 37 PAGES

This document reports the results of an analysis of the ap-
plicability of software specification and design languages
to the development of flight dynamics software at GSFC. Two
language systems were studied: HOS/AXES and PSL/PSA. The
approach was to develop software designs, using each of
these language systems, from the requirements of an already-
implemented system. The results of the two methodologies
were then compared with each other and with the actual de-
sign implemented. This document was also issued as a Martin
Marietta Corporation technical report. NSTF

5.8 GSFC SOFTWARE ENGINEERING RESEARCH REQUIREMENTS ANALYSIS STUDY, SEL-78-006, P. A. SCHEFFER AND C. E. VELEZ, NOVEMBER 1978, 26 PAGES

This document reports the results of a study of the applicability of requirements languages to flight dynamics software development at GSFC. The specific objectives of the study, which are explained in this document, were to

- Determine the impact of requirements language use on software design

- Demonstrate the application of a requirements language on a flight dynamics development problem

- Evaluate the utility of the Multi-Level Expression Design Language - Requirement Level (MEDL-R) in the GSFC environment

- Determine the desirable characteristics of a requirements language tool for use in the GSFC environment

This document was also issued as a Martin Marietta Corporation technical memorandum. NSTF

8978

5.9 MULTI-LEVEL EXPRESSION DESIGN LANGUAGE - REQUIREMENT LEVEL (MEDL-R) SYSTEM EVALUATION, SEL-80-002, W. J. DECKER AND C. E. GOOREVICH, MAY 1980, 91 PAGES

This document presents the results of an evaluation of the suitability of the Multi-Level Expression Design Language - Requirement Level (MEDL-R) for use in flight dynamics software development at GSFC. The evaluation team studied the MEDL-R concept of requirements languages, the functions performed by MEDL-R, and the MEDL-R language syntax. The document contains recommendations for changes to the MEDL-R system that would make it more useful in the flight dynamics environment. This document was also issued as Computer Sciences Corporation document CSC/TM-80/6093. NSTF

8978

5.10 <u>MULTI-MISSION MODULAR SPACECRAFT GROUND SUPPORT SOFT-</u>
<u>WARE SYSTEM (MMS/GSSS) STATE-OF-THE-ART COMPUTER</u>
<u>SYSTEMS/COMPATIBILITY STUDY</u>, SEL-80-003, T. WELDEN,
M. MCCLELLAN, AND P. LIEBERTZ, MAY 1980, 66 PAGES

This document describes the results of an evaluation of the compatibility between the ModComp IV/35 and the VAX-11/780 computers with respect to a specific software system, the Multi-Mission Modular Spacecraft Ground Support Software System (MMS/GSSS). The degree of compatibility was measured by comparing the results of benchmark tests run on both systems. The tests examined input/output services, FORTRAN language implementation, and execution timing. The compatibility of peripheral devices and system command languages was considered in lesser detail. Significant incompatibilities were found in all areas examined. This document was also issued as Computer Sciences Corporation document CSC/TM-80/6154. <u>NTIS</u>

8978

5.11 RECOMMENDED APPROACH TO SOFTWARE DEVELOPMENT, SEL-81-205, F. E. MCGARRY, G. PAGE, S. ESLINGER, ET AL., APRIL 1983, 278 PAGES

This document presents recommendations for a disciplined approach to software development, based on data collected and studied by the SEL since 1977 for approximately 40 flight dynamics software projects. It describes the major activities, end products, methodologies, tools, models, and measures applicable to each phase of the software life cycle. Life cycle phases include requirements analysis, preliminary design, detailed design, implementation, system testing, and acceptance testing.

The emphasis of the document is on management considerations, including the activities specific to each phase of the life cycle and the general management concerns transcending phase boundaries, such as development planning and performance monitoring. Specific recommendations are made with respect to determining project quality.

This version of the document contains new appendixes covering the topics of software reviews, development documents, examples of steps to organize a project, and a summary of key information. This document was also issued as Computer Sciences Corporation document CSC/TM-83/6019. NTIS

The previous versions of this document were Standard Approach to Software Development, SEL-81-005, V. E. Church, F. E. McGarry, and G. Page, September 1981, and Recommended Approach to Software Development, SEL-81-105, S. Eslinger, F. E. McGarry, and G. Page, May 1982.

5-12

5.12 "SOFTWARE ENGINEERING COURSE EVALUATION," G. PAGE, COMPUTER SCIENCES CORPORATION, TECHNICAL MEMORANDUM, DECEMBER 1977, 41 PAGES

This technical memorandum reproduces the results of an evaluation by students participating in a software engineering methods course. The students were actively working on software development projects at the time the course was offered. The objective of the evaluation was to determine the students' reaction to, previous familiarity with, and subsequent use of software engineering techniques. The memorandum includes the evaluation forms completed by the students and a tabulation of the responses. FEM

5.13 SOFTWARE ENGINEERING LABORATORY PROGRAMMER WORKBENCH PHASE 1 EVALUATION, SEL-81-009, W. J. DECKER AND F. E. MCGARRY, MARCH 1981, 29 PAGES

This document summarizes an initial effort to develop a programmer workbench for flight dynamics software development activities. Phase 1 of the programmer workbench consists of the design of three components: the communications link, the command language processor, and the software tools package. The document also contains a brief description and evaluation of the design of each component. Some recommendations for future work are made. This document was also issued as Computer Sciences Corporation document CSC/TM-81/6091. NSTF

8978

5.14 "USE OF CLUSTER ANALYSIS TO EVALUATE SOFTWARE
      ENGINEERING METHODOLOGIES," E. CHEN AND
      M. V. ZELKOWITZ, PROCEEDINGS OF THE FIFTH INTERNATIONAL
      CONFERENCE ON SOFTWARE ENGINEERING. NEW YORK:
      COMPUTER SOCIETIES PRESS, 1981, 7 PAGES

This technical paper describes an attempt to identify the
characteristic effects of various methodologies on software
development. Data collected by the SEL from five software
projects was studied. Several objective measures were de-
rived from the data, and their relationships to methodology
use were studied with cluster analysis techniques. The
analysis showed that the measures reflected the effects of
methodologies on software development. JAO

This technical paper also appears in SEL-82-004, Collected
Software Engineering Papers: Volume I, July 1982.

8978

5.15 "A SOFTWARE ENGINEERING VIEW OF THE FLIGHT DYNAMICS
ANALYSIS SYSTEM (FDAS): PARTS I AND II," D. N. CARD,
W. W. AGRESTI, V. E. CHURCH, AND Q. L. JORDAN, COMPUTER
SCIENCES CORPORATION, TECHNICAL MEMORANDUM,
DECEMBER 1983 (PART I) AND MARCH 1984 (PART II),
58 PAGES

This report presents the results of an assessment, from the
software engineering point of view, of the Flight Dynamics
Analysis System (FDAS) at one step in the requirements def-
inition process--a prototype support environment. FDAS is
intended to provide an integrated software development sup-
port environment for research applications in the areas of
orbit, attitude, and mission analysis, and it was conceived
to assist users in the preparation, execution, and interpre-
tation of software experiments. A prototype FDAS was con-
structed to aid in clarifying the requirements for such a
system and to test some concepts of language, software
structure, and user interface designs. Part I of the report
discusses the general approaches to FDAS adapted by the de-
velopment team. Part II presents a detailed examination of
some high-level FDAS design issues and summarizes some simi-
lar systems from other environments.

8978

5.16 "A PRACTICAL EXPERIENCE WITH INDEPENDENT VERIFICATION
      AND VALIDATION," G. PAGE, F. E. MCGARRY, AND
      D. N. CARD, PROCEEDINGS OF THE EIGHTH INTERNATIONAL
      COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE,
      NOVEMBER 1984, 5 PAGES

This paper describes an attempt to assess the benefits and
limitations of the application of independent verification
and validation (IV&V) in the flight dynamics area at NASA/
GSFC. The SEL applied the IV&V methodology to two medium-
sized flight dynamics software development projects. Then,
to measure the effectiveness of the IV&V approach, the SEL
compared these two projects with two similar past projects,
using measures like productivity, reliability, and maintain-
ability. Results indicated that the use of the IV&V method-
ology did not help the overall process nor improve the
product in these cases. JAO

8978

5.17 "QUANTITATIVE EVALUATION OF SOFTWARE METHODOLOGY,"
V. R. BASILI, PROCEEDINGS OF THE FIRST PAN-PACIFIC
COMPUTER CONFERENCE, SEPTEMBER 1985, 21 PAGES

This paper presents a paradigm for evaluating software de-
velopment methods and tools. The basic idea is to generate
a set of goals that are refined into quantifiable ques-
tions. These questions specify the metrics to be collected
on the software development and maintenance process and pro-
duct. The metrics can be used to characterize, evaluate,
predict, and motivate. They can be used in an active as
well as passive way by learning from analyzing the data and
improving the methods and tools based on what is learned
from that analysis. Several examples were given repre-
senting each of the different approaches to evaluation. JAO

8978

5.18 "FOUR APPLICATIONS OF A SOFTWARE DATA COLLECTION AND ANALYSIS METHODOLOGY," V. R. BASILI AND R. W. SELBY, JR., PROCEEDINGS OF THE NATO ADVANCED STUDY INSTITUTE, AUGUST 1985, 15 PAGES

This paper presents a seven-step data collection and analysis methodology that couples software technology evaluation with software measurement. Four in-depth applications of the methodology are presented. The four studies represent each of the general categories of analyses on the software product and development process: blocked subject-project studies, replicated project studies, multiproject variation studies, and single project studies. The four applications are in the areas of software testing strategies, Cleanroom software development, characteristic software metric sets, and software error analysis, respectively. JAO

8978

5.19 "MEASURING THE IMPACT OF COMPUTER RESOURCE QUALITY ON
      THE SOFTWARE DEVELOPMENT PROCESS AND PRODUCT,"
      F. E. MCGARRY, J. VALETT, AND D. HALL, PROCEEDINGS
      OF THE HAWAIIAN INTERNATIONAL CONFERENCE OF SYSTEM
      SCIENCES, JANUARY 1985, 9 PAGES

This study examined the relationship between computer re-
sources and the software development process and product as
exemplified by NASA/GSFC data. Data have been extracted and
examined from nearly 50 software development projects
varying in size from 3,000 to 130,000 lines of code. All
have been related to the support of satellite flight
dynamics ground-based computations. As a result of changing
situations and technology, the computer support environment
has varied widely. Some projects enjoyed fast response
time, excess memory, and state-of-the-art tools, whereas
others endured slow computer response time, archaic tool
support, and limited terminal access to the development
machine. Based on the results of this study, a number of
computer-resource-related implications are provided. JAO

8978

5.20 "ANALYZING THE TEST PROCESS USING STRUCTURAL COVERAGE," J. RAMSEY and V. R. BASILI, PROCEEDINGS OF THE EIGHTH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, AUGUST 1985, 7 PAGES

This paper reports the results of a study to understand and improve the acceptance test process in the SEL environment. An SEL program, the MAL language preprocessor (a subset of a satellite attitude maintenance system), has been modified to produce structural coverage metrics. It was modified to measure both procedure coverage and statement coverage. Coverage is also computed for several statement subclasses. The modified program was executed on a set of functionally generated acceptance tests and a large sample of operational usage cases. The resulting structural coverage metrics are combined with fault and error data to evaluate structural coverage in the SEL environment.

It is shown that, in this environment, the functionally generated tests seem to be a good approximation of operational use. The relative proportions of the exercised statement subclasses change as the structural coverage of the program increases. A method is proposed for evaluating whether two sets of input data exercise a program in a similar manner. Evidence also shows that (1) faults revealed in a procedure are independent of the number of times the procedure is executed and (2) it may be reasonable to use procedure coverage in software models that use statement coverage. Finally, the evidence suggests that it may be possible to use structural coverage to aid in managing the acceptance test process. JAO

This technical paper also appears in SEL-85-003, Collected Software Engineering Papers: Volume III, November 1985, and in Structural Coverage of Functional Testing, University of Maryland Technical Report, TR-1442, September 1984.

8978

5.21 EVALUATIONS OF SOFTWARE TECHNOLOGIES: TESTING, CLEAN-ROOM, AND METRICS, SEL-85-004, R. W. SELBY, JR., MAY 1985, 183 PAGES

This document describes a seven-step approach for quantitatively evaluating software technologies coupling software methodology evaluation with software measurement. The approach is applied in depth in the following three areas:

- Software Testing Strategies -- A 74-subject study, including 32 professional programmers and 42 advanced university students, compared code reading, functional testing, and structural testing in a fractional factorial design.

- Cleanroom Software Development -- Fifteen 3-person teams separately built a 1200-line message system to compare Cleanroom software development (in which software is developed completely off line) with a more traditional approach.

- Characteristic Software Metric Sets -- In the SEL production environment, a study of 65 candidate product and process measures of 652 modules from 6 projects of 51,000 to 112,000 lines yielded a characteristic set of software cost/quality metrics. FEM

This technical paper was also issued as University of Maryland Technical Report, TR-1500, May 1985.

8978

5.22 COMPARISON OF SOFTWARE VERIFICATION TECHNIQUES, SEL-85-001, D. N. CARD, R. W. SELBY, F. E. MCGARRY, ET AL., APRIL 1985, 90 PAGES

This document describes a controlled experiment performed by the SEL to compare the effectiveness of code reading, functional testing, and structural testing as software verification techniques. It is one of a series of three experiments organized by R. W. Selby as part of his doctoral dissertation. The experiment results indicate that code reading provides the greatest error detection capability at the lowest cost, whereas structural testing is the least effective technique. This document explains the experiment plan, describes the experiment results, and discusses related results from other studies. It also considers the application of these results to the development of software in the flight dynamics environment. Appendixes summarize the experiment data and list the test programs. A separate Data Supplement contains original materials collected from the participants. NTIS

5-23

5.23 ADA TRAINING EVALUATION AND RECOMMENDATIONS FROM THE
      GAMMA RAY OBSERVATORY ADA DEVELOPMENT TEAM, SEL-85-002,
      R. MURPHY AND M. STARK, OCTOBER 1985, 41 PAGES

This document presents the Ada training experiences of the
Gamma Ray Observatory Ada development team and recommenda-
tions for future Ada training of software developers.
Training methods are evaluated; deficiencies in the training
program are noted; and a recommended approach, including
course outline, time allocation, and reference materials, is
offered. NTIS

SECTION 6 - DATA COLLECTION

6.1 A COMPARISON OF RADC AND NASA/SEL SOFTWARE DEVELOPMENT DATA, C. TURNER AND G. CARON, DATA & ANALYSIS CENTER FOR SOFTWARE, MAY 1981, 31 PAGES

This document reports the results of an analysis of the relationship between project size and several other software measures. These measures include productivity, effort, duration, errors, and error rate. The analysis used data from two sources: Rome Air Development Center (RADC) and the SEL. Least-squares regression techniques were applied to both sets of data. Results obtained from the two sets of data were comparable. The conclusion cited in the report is that RADC and SEL data can be combined, in most cases, to obtain a larger sample without undesirable side effects.

6.2 A METHODOLOGY FOR COLLECTING VALID SOFTWARE ENGINEERING DATA, V. R. BASILI AND D. M. WEISS, UNIVERSITY OF MARYLAND, TECHNICAL REPORT, DECEMBER 1982, 22 PAGES

This technical report describes an effective data collection method for evaluating software methodologies and for studying the software development process. The purpose of the report is to show how to obtain valid data that may be used both to learn more about the software development process and to evaluate software development methodologies in a production environment. The data collected during the study describes changes made to the software during development and is obtained when the changes are made. To ensure accuracy of the data, validation is performed concurrently with software development as part of the data collection process. Validation is based on interviews with the programmers supplying the data. The feasibility of the data collection methodology was demonstrated by applying it to five difference project in two different production environments. FEM

This technical paper also appears in SEL-83-003, Collected Software Engineering Papers: Volume II, November 1983.

6.3 AUTOMATED COLLECTION OF SOFTWARE ENGINEERING DATA IN THE SOFTWARE ENGINEERING LABORATORY (SEL), SEL-81-014, A. L. GREEN, W. J. DECKER, AND F. E. McGARRY, SEPTEMBER 1981, 72 PAGES

This document presents the results of an analysis of SEL data collection procedures. The principal questions addressed are what current manual procedures could be automated and how these automated procedures could be incorporated in the SEL data base system. The functional requirements of such a system are identified and explained. The automatable sources of data identified in this report include the following:

- Computer accounting information
- Requirements language tools
- Program design language tools
- Programmer workbench features
- Source code analyzer program

This document was also issued as Computer Sciences Corporation document CSC/TM-81/6222. NSTF

6.4 "CONCEPTS USED IN THE CHANGE REPORT FORM," F. PARR AND
    D. M. WEISS, NASA, GODDARD SPACE FLIGHT CENTER,
    TECHNICAL MEMORANDUM, MAY 1978, 9 PAGES

This technical memorandum discusses the software engineering
basis of the Change Report Form devised by the SEL. This
form is used to collect change and error data from flight
dynamics software development projects at GSFC. The memo-
randum describes the questions that appear on the form and
explains the possible answers. These questions attempt to
determine the nature of the change/error, the cause of the
change/error, the cost of the change/error, and other re-
lated qualities. FEM

6.5 "DATA COLLECTION AND EVALUATION FOR EXPERIMENTAL
COMPUTER SCIENCE RESEARCH," M. V. ZELKOWITZ, EMPIRICAL
FOUNDATIONS FOR COMPUTER AND INFORMATION SCIENCE
(PROCEEDINGS), NOVEMBER 1982, 15 PAGES

This technical paper reviews the data collection procedures
of the SEL and shows how they are used to generate data with
one particular software engineering experiment. The SEL
collects process and product measures from actual scientific
software projects using a combination of automated tools and
questionnaires. A project currently in progress, from which
data is being collected, is a software prototyping effort.
The goal of this experiment is to determine the costs and
benefits of developing a prototype before developing a full
system. JAO

8978

6.6 EVALUATING SOFTWARE DEVELOPMENT BY ANALYSIS OF CHANGES: THE DATA FROM THE SOFTWARE ENGINEERING LABORATORY, SEL-82-008, V. R. BASILI AND D. M. WEISS, DECEMBER 1982, 77 PAGES

This document reports the results of a study for evaluating software development by analyzing changes to the software. The specific goals of the study were to

- Characterize changes and errors

- Characterize projects and programmers

- Identify effective error detection and correction techniques

- Investigate ripple effects in the software caused by changes

The data collected for the report consisted of changes (including error corrections) made to the software after code was written and baselined, but before testing began. Data collection and validation were concurrent with software development. Changes reported were verified by interviews with the originating programmers. Analysis of the data used in the study showed patterns that were used in satisfying the goals of the data collection. (Also see Section 5.3.) A version of this document was also issued as University of Maryland Technical Report TR-1236. NTIS

8978

6.7 GUIDE TO DATA COLLECTION, SEL-81-101, V. E. CHURCH, D. N. CARD, AND F. E. McGARRY, AUGUST 1982, 123 PAGES

This document presents guidelines and recommendations for collecting software development data. The guide describes the motivation, planning, implementation, and management of a data collection effort. Other topics covered include types, sources, and availability of data; methods and costs of data collection; types of analyses supported; and warnings and suggestions based on SEL experience. The appendixes include facsimiles of SEL data collection forms and a glossary of software engineering terms.

This document, abstracted and generalized from 5 years of SEL data collection experience, is intended to be a practical guide for software managers and engineers. It was also issued as Computer Sciences Corporation document CSC/TM-82/6137. NTIS

The previous version of this document was Guide to Data Collection, SEL-81-001, V. E. Church, D. N. Card, and F. E. McGarry, September 1981.

8978

6.8 NASA/SEL DATA COMPENDIUM, C. TURNER, G. CARON, AND G. BREMENT, DATA & ANALYSIS CENTER FOR SOFTWARE, SPECIAL PUBLICATION, MAY 1981, 80 PAGES

This document summarizes the software engineering data collected by the SEL from flight dynamics development projects. A series of charts and graphs is presented for each project. These materials provide profiles of the development histories of the projects and define some of the major characteristics of the software development process at GSFC. However, this document is a factual presentation rather than an analysis of the data.

8978

6.9 "OPERATION OF THE SOFTWARE ENGINEERING LABORATORY,"
V. R. BASILI AND M. V. ZELKOWITZ, PROCEEDINGS OF THE
SECOND SOFTWARE LIFE CYCLE MANAGEMENT WORKSHOP,
AUGUST 1978, 4 PAGES

This technical paper describes the operation of the SEL.
Software engineering data is regularly collected by the SEL
from flight dynamics software development projects at GSFC.
The assembled data supports an extensive program of software
engineering research. This report also reviews SEL data
collection and data processing activities and their rela-
tionship to the research program. It also summarizes some
ongoing resource estimation and error analysis research
projects. JAO

This technical paper also appears in SEL-82-004, Collected
Software Engineering Papers: Volume I, July 1982.

8978

6.10 "OPERATIONAL ASPECTS OF A SOFTWARE MEASUREMENT
       FACILITY," M. V. ZELKOWITZ AND V. R. BASILI,
       PROCEEDINGS OF THE SOFTWARE LIFE CYCLE MANAGEMENT
       WORKSHOP, SEPTEMBER 1977, 11 PAGES

This technical paper describes the data collection opera-
tions of the SEL. The source of the data is a group of
flight dynamics software development projects at GSFC. The
paper describes seven data collection forms used by the
SEL. The procedure for transferring data from the forms to
the computer data base is outlined. Some of the validity
checks performed on the data are identified. This procedure
produces valid, relevant data with which significant soft-
ware engineering research can be conducted. JAO

6.11 "A METHODOLOGY FOR COLLECTING VALID SOFTWARE ENGINEER-
ING DATA," V. R. BASILI AND D. M. WEISS, <u>IEEE TRANS-
ACTIONS ON SOFTWARE ENGINEERING</u>, NOVEMBER 1984,
II PAGES

This paper describes an effective data collection method for
evaluating software development methodologies and studying
the software development process. The method uses
goal-directed data collection to evaluate methodologies with
respect to the claims made for them. Such claims are used
as a basis for defining the goals of the data collection,
establishing a list of questions of interest to be answered
by data analysis, defining a set of data categorization
schemes, and designing a data collection form.

The data to be collected are based on the changes made to
the software during development and are obtained when the
changes are made. To ensure accuracy of the data,
validation is performed concurrently with software
development and data collection. Validation is based on
interviews with those people supplying the data. Results
from using the methodology show that data validation is a
necessary part of change data collection. Without it, as
much as 50 percent of the data may be erroneous.

Feasibility of the data collection methodology was
demonstrated by applying it to five different projects in
two different environments. The application showed that the
methodology was both feasible and useful. <u>JAO</u>

8978

INDEX OF SUBJECTS

8978

implementation,

    2.3, 4.18, 4.40, 5.11

life cycle,

    4.3, 4.18, 5.11

McCabe measures,

    3.8, 4.1, 4.15, 4.20, 4.27

measures,

    2.1, 2.3, 2.7, 2.8, 2.9, 2.10, 2.11, 2.14, 2.16, 3.6,
    3.8, 4.1, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19, 4.20,
    4.21, 4.22, 4.23, 4.24, 4.27, 4.29, 4.30, 4.32, 4.33,
    4.34, 4.36, 4.37, 4.40, 4.41, 4.42, 5.2, 5.17, 5.19,
    5.20, 6.1, 6.8

methodologies,

    2.1, 2.3, 2.5, 2.8, 2.9, 2.14, 3.2, 4.16, 4.18, 4.26,
    4.35, 4.39, 4.40, 5.3, 5.4, 5.7, 5.11, 5.12, 5.14, 5.15,
    5.16, 5.17, 5.21

metrics,

    see measures

models,

    see reliability models, resource models

portability

    2.11, 3.7, 5.10

productivity,

    see factors, resource models

program design languages,

    2.6, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9

8978

reliability models,

2.8, 2.10, 2.11, 2.14, 2.16, 3.5, 4.4, 4.5, 4.13, 4.14,
4.18, 4.22, 4.23, 4.28, 5.2, 5.3, 5.12, 6.4

requirements analysis,

2.6, 2.14, 4.36, 5.8, 5.9, 5.11

resource models,

2.1, 2.3, 2.8, 2.9, 2.10, 2.11, 2.16, 3.5, 4.2, 4.3,
4.6, 4.8, 4.9, 4.12, 4.14, 4.18, 4.22, 4.23, 4.24, 4.25,
4.26, 4.31, 4.32, 4.41, 4.42, 5.19

Software Engineering Laboratory,

2.1, 2.2, 2.4, 2.5, 2.6, 2,7, 2.8, 2.9, 2.10, 2.11,
2.12, 2.13, 2.15, 2.16, 2.17, 4.7, 4.20, 6.3, 6.5, 6.9,
6.10

Software Engineering Laboratory data base,

2.7, 2.8, 2.15, 2.16, 2.18, 3.10, 3.12, 3.13, 3.14,
3.18, 3.19, 6.8, 6.9

Software Engineering Laboratory terms,

2.4

Software Engineering Workshop,

2.5, 2.6, 2.7, 2.8, 2.9, 2.10, 2.11, 2.12, 2.13, 2.14

statistical methods,

2.7, 2.11, 4.16, 5.14

testing,

2.3, 2.13, 2.14, 4.18, 4.38, 5.4, 5.11, 5.16, 5.20,
5.21, 5.22

8978

tools,

2.3, 2.5, 2.6, 2.14, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.20, 4.18, 5.5, 5.6, 5.7, 5.8, 5.9, 5.13, 5.23

8978